A spectral approach for characterizing the self-synchronization of stream ciphers

Jeremy Parriaux, Philippe Guillot, Gilles Millérioux

A Spectral Approach for Characterizing the Self-Synchronization of Stream Ciphers

Jérémy Parriaux[1], Philippe Guillot[2], and Gilles Millérioux[1]

[1] Nancy University, CNRS,
Research Center for Automatic Control of Nancy (CRAN UMR 7039), France,
jeremy.parriaux@esstin.uhp-nancy.fr, gilles.millerioux@esstin.uhp-nancy.fr,
[2] Université Paris 8
Laboratoire Analyse, Géométrie et Applications (LAGA UMR 7539), France
philippe.guillot@univ-paris8.fr

Abstract. This paper addresses the problem of characterizing the functions that can be used in the design of self-synchronizing stream ciphers. We propose a general framework based on a spectral characterization through the Walsh transform. Two modes of self-synchronization are discussed: the finite time one and the statistical one.

1 Introduction

Stream ciphers are cryptosystems specifically devoted to the transmission of data streams over public channels. The basic principle can be described as follows. At the transmitter side, the ciphertext is carried out by adding a plaintext symbol with a symbol of a pseudorandom stream. At the receiver side, the decryption consists in subtracting the ciphertext symbol with a symbol of, again, a pseudorandom stream. Proper decryption is achieved provided that the pseudorandom streams generated at the transmitter and receiver sides are the same. In other words, the pseudorandom generators have to be synchronized. There are two ways to ensure the synchronization. The first one is to use an external protocol in order to initialize the two generators with the same seed. The protocol must also be able to resynchronize the generators if the synchronization is lost. The resulting ciphers are known as synchronous stream ciphers. The second method relies on systems for which synchronization is due to a structural property. The corresponding ciphers are called self-synchronizing stream ciphers (SSSC for short). The absence of synchronization protocol makes them particularly appealing when high throughputs are required. As it turns out, very few works have paid attention to them. Let us mention [1,2] for an exception. This work aims at characterizing new functions which can be involved in self-synchronizing stream ciphers. The interest of enlarging the class of candidates functions lies in that they can potentially lead to systems of reduced size or with better cryptographic properties than the existing ones. The characterization is performed in the spectral domain and thereby allows to connect the results to the usual cryptographic criteria.

The outline of this paper is the following: Section 2 is devoted to the problem statement. Section 3 recalls the usual material devoted to spectral analysis and Boolean functions. Section 4 deals with the spectral characterization of the self-synchronizing property and is the core of the paper. Section 5 investigates the reachability of the states in terms of probability law. Finally, Section 6 is devoted to an illustrative example.

2 Problem Statement

Let us first introduce the notations. The two elements field is denoted $\mathbb{F}_2$. The plaintext symbol to be ciphered at time $t \in \mathbb{N}$ is $m_t \in \mathbb{F}_2$, the corresponding ciphertext is $c_t \in \mathbb{F}_2$ and the corresponding recovered plaintext is $\widehat{m}_t \in \mathbb{F}_2$. In stream-ciphers, the ciphertext c_t is obtained

from the plaintext m_t by adding a random symbol $z_t \in \mathbb{F}_2$. The original message $\widehat{m}_t$ is recovered by subtracting the symbol $\widehat{z}_t \in \mathbb{F}_2$ from the ciphertext c_t. In the canonical representation of a self-synchronizing stream cipher, the random symbols z_t and $\widehat{z}_t$ are generated using the same keyed function $g_\theta : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ whose arguments are a finite sequence of some past ciphertexts, namely $c_{t-1}, \ldots, c_{t-n}$. The parameter θ is the key of the system. The decryption is properly performed, that is $\widehat{m}_t = m_t$, whenever $\widehat{z}_t = z_t$. It is guaranteed if both the cipher and the decipher have the same key and if the ciphertexts $c_{t-1}, \ldots, c_{t-n}$ are properly transmitted. The equations of the canonical form of self-synchronizing stream ciphers are

$$\begin{cases} z_t = g_\theta(c_{t-1}, \ldots, c_{t-n}) \\ c_t = m_t + z_t \end{cases} \text{(Cipher equation)} \tag{1}$$

$$\begin{cases} \widehat{z}_t \ = g_\theta(c_{t-1}, \ldots, c_{t-n}) \\ \widehat{m}_t = c_t - \widehat{z}_t \end{cases} \text{(Decipher equation)} \tag{2}$$

The canonical form admits an equivalent recursive form involving an internal state $x \in \mathbb{F}_2^n$ which is an n–dimensional Boolean vector. Its value at time t is $x_t = (c_{t-1}, \ldots, c_{t-n})$. Its i^{th} coordinate is denoted $(x_t)_i$. The corresponding block diagram is depicted in Figure 1. The equations read

$$\begin{cases} (x_{t+1})_i = (x_t)_{i-1} \ \text{ if } i > 0, \ c_t \text{ if } i = 0 \\ z_t \quad\quad = g_\theta(x_t) \\ c_t \quad\quad = m_t + z_t \end{cases} \text{(Cipher equation)} \tag{3}$$

$$\begin{cases} (\widehat{x}_{t+1})_i = (\widehat{x}_t)_{i-1} \ \text{ if } i > 0, \ c_t \text{ if } i = 0 \\ \widehat{z}_t \quad\quad = g_\theta(\widehat{x}_t) \\ \widehat{m}_t \quad\quad = c_t - \widehat{z}_t \end{cases} \text{(Decipher equation)} \tag{4}$$

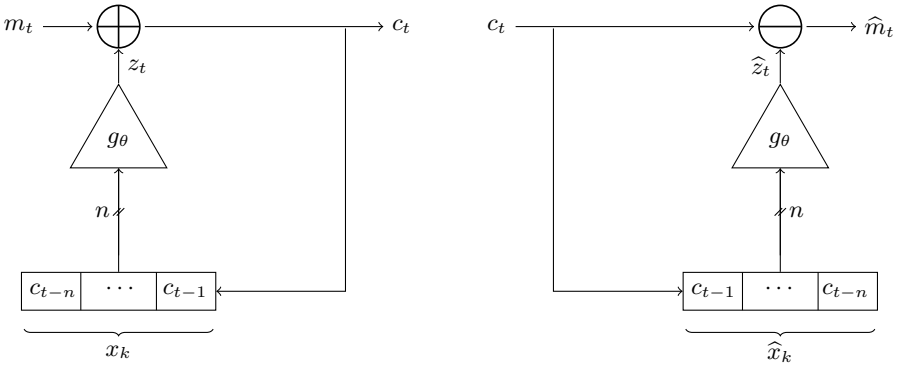

Fig. 1: Canonical recursive form of self-synchronizing stream ciphers

The canonical recursive form (3)–(4) is directly obtained from the canonical form (1)–(2). The state updating transformation is a mere shift register fed with the previous ciphertexts. Thus, the initial state is eliminated in a shift-like way and all the complexity of the system lies in the function g_θ. More interesting schemes are obtained when considering a keyed state updating transformation $f_\theta : \mathbb{F}_2 \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ more complex than a shift. In this situation, the shift next state function and the output function g_θ are replaced by a function f_θ and an output

function $h_\theta : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. This setup is referred to as the generalized recursive form and its block diagram is depicted in Figure 2. The corresponding equations are

$$\begin{cases} x_{t+1} = f_\theta(c_t, x_t) \\ z_t \quad = h_\theta(x_t) \quad \text{(Cipher equation)} \\ c_t \quad = m_t + z_t \end{cases} \tag{5}$$

$$\begin{cases} \widehat{x}_{t+1} = f_\theta(c_t, \widehat{x}_t) \\ \widehat{z}_t \quad = h_\theta(\widehat{x}_t) \quad \text{(Decipher equation)} \\ \widehat{m}_t \quad = c_t - \widehat{z}_t \end{cases} \tag{6}$$

In order to guarantee the self-synchronization property of the system, the function f_θ cannot be chosen arbitrarily. It must have the property that, after a fixed number, denoted t_c, of iterations, the pseudorandom symbols z_t and $\widehat{z}_t$ are equal for all $t > t_c$. In the general case, this is achieved if and only if the current state of the decipher is equal to the current state of the cipher, $\widehat{x}_t = x_t$ regardless of the initial states x_0 and $\widehat{x}_0$. Clearly, given the system described by (5)–(6), the self-synchronization can be studied by focusing exclusively on the function f_θ. Besides, the fact that this recursive form is more general than a mere shift, it allows to relax the constraint that the synchronization is achieved within a finite amount of time. That leads to so-called statistical self-synchronizing stream ciphers. They will be detailed and motivated later on in this paper.

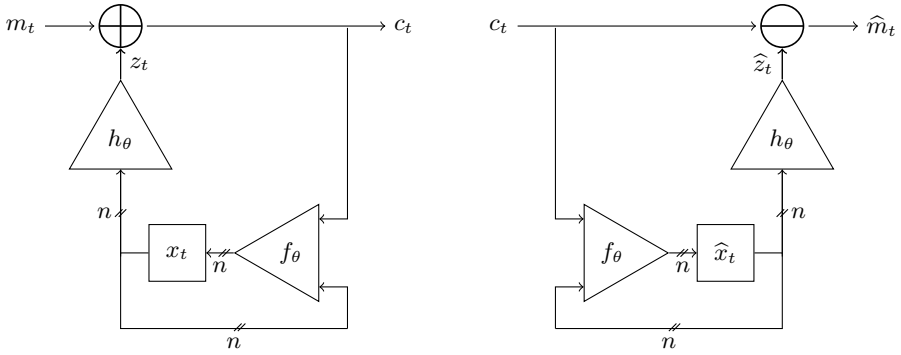

Fig. 2: Generalized recursive form of self-synchronizing stream ciphers

This paper does not intend to study how the key θ is involved in the system. Therefore, for simplification purposes and hereafter, the subscript θ will be omitted, the parameterization with the key of the functions will be implicit. In the sequel, a function from the vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2$ will be referred to as a (n)–function. We will call a (n, m)–function a function from the vector space $\mathbb{F}_2^n$ to the vector space $\mathbb{F}_2^m$. Indeed, a (n, m)–function f is nothing but a m–dimensional vector where each coordinate is a (n)–function. The j^{th} coordinate is denoted by f_j and named the coordinate function. We recall a definition introduced by Klimov and Shamir in [3].

Definition 1 (T–function). *A (n, n)–function is called a T–function if the coordinate function f_j depends only on the variables x_i with $i = 0, \ldots, j$.*

Some special T–functions of interest in our study are called strict T–functions. Their definition is the following:

Definition 2 (Strict T–function). *A T–function such that the coordinate function f_j depends only on the variable x_i with $i = 0, \ldots, j-1$ is called a strict T–function.*

Remark 1. It should be mentioned than what we call *strict T–function* is nothing but what is called *parameter* in [3]. We however prefer not to use this name because it might be confusing in some situations.

Having a look at the literature, it can be noticed that, so far, all the self-synchronizing stream ciphers use the same principle in order to guarantee that the current state at time t no longer depends on the initial state, that is to guarantee the self-synchronization. The state updating function is such that its coordinate functions depend on the bits of the internal state with strictly lower indexes than their own index. In other words, the state updating function is based on strict T–functions.

The main purpose of this paper is to pinpoint more general classes of functions which guarantee the self-synchronization property besides the strict T–functions. Self-synchronization properties are addressed from a spectral point of view as motivated in the introduction.

3 Preliminaries

This section introduces a formal definition of self-synchronization and then recalls the strict necessary prerequisites on spectral analysis of Boolean functions from which our results will be derived.

3.1 Self-synchronization

Let us first formally define some self-synchronization related notions. In the paper, (c) is a sequence of the ciphertexts $c_0, \ldots, c_t$ for some discrete time t, they are generated by (5). Thus, the length of the sequence at time t is $t+1$ and according to (5), c_0 is only a function of the initial state x_0, they are related by $c_0 = h(x_0)$.

Definition 3 (Self-synchronizing sequence). *A sequence (c) is self-synchronizing for the next state $(n+1, n)$–function f of the system (5)–(6) if there exists an integer t_c so that for all initial states $x_0 \in \mathbb{F}_2^n$ and $\widehat{x}_0 \in \mathbb{F}_2^n$*

$$\forall t \geq t_c, \ x_t = \widehat{x}_t \tag{7}$$

Definition 4 (Finite time self-synchronization). *The system (5)–(6) is finite time self-synchronizing if the minimum value t_c is upper bounded for all possible sequences (c). The upper bound t_c is called the self-synchronization delay of f.*

Remark 2. Finite time self-synchronization means that there is an integer t_c such that any sequence of length at least t_c is a self-synchronizing sequence. The synchronization delay depends on the pair of initial states x_0 and $\widehat{x}_0$. The delay t_c is defined as the maximum delay over all initial state pairs.

Definition 5 (Finite time self-synchronizing function). *A $(n+1, n)$–function f is called finite time self-synchronizing function if, when used as a next state function in the system (5)–(6), the resulting system is finite time self-synchronizing.*

3.2 Spectral Analysis

The rest of this section recalls the basics about Boolean spectral analysis. If f is a (n)–function, we denote by $\widehat{f}$ its Fourier transform, which is by definition the real valued mapping $\mathbb{F}_2^n \longrightarrow \mathbb{R}$ defined, for any $u \in \mathbb{F}_2^n$, by

$$\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u} \tag{8}$$

where $x \cdot u = x_0 u_0 + \cdots + x_{n-1} u_{n-1}$. This transform is invertible and the inverse is given by:

$$\widehat{\widehat{f}} = 2^n f \tag{9}$$

Let us recall Parseval's theorem (see [4]):

Theorem 1 (Parseval's theorem). *For any Boolean function $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$, the following relation holds:*

$$\sum_{u \in \mathbb{F}_2^n} \widehat{f}(u)^2 = 2^n \sum_{x \in \mathbb{F}_2^n} f(x)^2 \tag{10}$$

When dealing with Boolean functions, we rather resort to the Walsh transform which gets nicer properties than the Fourier transform in most cases. The Walsh transform of a Boolean function f is the Fourier transform of its sign function f_χ where $f_\chi = (-1)^{f(x)} = 1 - 2f(x)$ that is,

$$\widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+x \cdot u} \tag{11}$$

As shown in [4], the correspondence between the Fourier and Walsh transforms is given by

$$\forall u \in \mathbb{F}_2^n, \quad \widehat{f_\chi}(u) = 2^n \delta_0(u) - 2\widehat{f}(u), \tag{12}$$

where $\delta_0(u)$ is the Kronecker symbol. It is equal to 1 if u is the n–dimensional zero vector and equals to 0 elsewhere.

The Walsh matrix of any (n, m)–function is the $2^m \times 2^n$ dimensional matrix $W_f = \left(w^f_{u,v}\right)$ so that (see [5]):

$$\forall u \in \mathbb{F}_2^m,\ \forall v \in \mathbb{F}_2^n,\ w^f_{u,v} = \sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot f(x)+v \cdot x} \tag{13}$$

The row $u \in \mathbb{F}_2^m$ of the matrix W_f is the Walsh transform of the linear combinations of the coordinates of f defined by $x \longmapsto u \cdot f(x)$. The coefficients of the Walsh matrix of a function is called the spectrum of that function.

N.B. Matrices indexes may be without ambiguity either an integer or a binary vector being the binary expansion of this integer.
An interesting property relates the Walsh matrices of composed functions.

Proposition 1 (see [5]). *If f is a (n, m)–function and g is a (p, n)–function then*

$$W_{f \circ g} = \frac{1}{2^n} W_f \times W_g \tag{14}$$

After these necessary preliminaries, we are now in a position to characterizing the self-synchronization property from a spectral point of view.

4 Spectral Characterization of the Self-Synchronization Property

In this section, we focus on characterizing the self-synchronizing property of the system (5)–(6). As motivated earlier, we can exclusively focus on the next-state function f. It is a $(n+1,n)$–function. Let us denote by f^0 (respectively f^1) the (n,n)–function which is the restriction of f to the input bit $c_t = 0$ (respectively to $c_t = 1$). The function f can be written as

$$f(c_t, x_t) = \begin{cases} f^0(x_t) & \text{if } c_t = 0 \\ f^1(x_t) & \text{if } c_t = 1 \end{cases} \tag{15}$$

For our purpose, we must define the function ϕ_t, the t^{th} order iterated function of f. It is the $(n+t+1,n)$–function defined by

$$\phi_t(c, x_0) = f^{c_t} \circ \cdots \circ f^{c_0}(x_0) \text{ for } t > 0$$

with $\phi_0 = f$. For a prescribed ciphertext sequence (c) of length $t+1$ and an initial state x_0, the function ϕ_t delivers the value of the internal state at time $t+1$.

In this section, we characterize the self-synchronizing property in terms of Walsh coefficients. We first focus on self-synchronizing sequences and then apply their properties to address the finite time and statistical self-synchronization issues of the system (5)–(6).

4.1 Self-synchronizing sequences

Let us denote by $\phi_t^c(x)$ the (n,n)–function which is the restriction of $\phi_t(c,x)$ to the specific sequence (c) of length $t+1$.

Proposition 2. *The sequence (c) is self-synchronizing if and only if the Walsh matrix of ϕ_t^c is a $2^n \times 2^n$ Walsh matrix of the form*

$$W_{\phi_t^c} = \begin{pmatrix} 2^n & 0 & \cdots & 0 \\ \pm 2^n & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \pm 2^n & 0 & \cdots & 0 \end{pmatrix} \tag{16}$$

Proof. By definition, if (c) is a self-synchronizing sequence, $\phi_t^c(x)$ does not depend on x thus, ϕ_t^c is a constant function. And yet, any linear combination of the coordinate functions of ϕ_t^c is also a constant function. It turns out that any row of (16) is the Walsh transform of a constant function. The converse can be derived by using the inverse Fourier transform formula (9).

The matrix $W_{\phi_t^c}$ can easily be determined from the knowledge of the Walsh matrices of f^0 and f^1.

Proposition 3. *The expression of the Walsh matrix $W_{\phi_t^c}$ is*

$$W_{\phi_t^c} = \frac{1}{2^{n \cdot t}} W_{f^{c_t}} \times \cdots \times W_{f^{c_0}} \tag{17}$$

Proof. The proof is a direct consequence of Proposition 1.

In the following, we use these results to derive some characteristics of the Walsh matrices of the functions that have the self-synchronization property.

4.2 Finite Time Self-Synchronization

Let us first notice some important features of Walsh matrices. In the sequel, we consider W as a square Walsh matrix of dimension $q \times q$ with $q = 2^n$.

$$W = \begin{pmatrix} q & 0 & \cdots & 0 \\ w_{1,0} & w_{1,1} & \cdots & w_{1,q-1} \\ \vdots & \vdots & & \vdots \\ w_{q-1,0} & w_{q-1,1} & \cdots & w_{q-1,q-1} \end{pmatrix} \tag{18}$$

The matrix W can be rewritten $W = A + N$ with

$$A = \begin{pmatrix} q & 0 \cdots 0 \\ w_{1,0} & 0 \cdots 0 \\ \vdots & \vdots \quad \vdots \\ w_{q-1,0} & 0 \cdots 0 \end{pmatrix} \qquad N = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & w_{1,1} & \cdots & w_{1,q-1} \\ \vdots & \vdots & & \vdots \\ 0 & w_{q-1,1} & \cdots & w_{q-1,q-1} \end{pmatrix}$$

A matrix is said to be of type A if the only non zero coefficients are located on the first column. A matrix is said to be of type N if all the coefficients of the first row and first column are zero. It is straightforward to verify the following remark:

Remark 3.
- the product of any two matrices of type A is a matrix of type A;
- the product of any matrix of type A with any matrix of type N is a zero matrix;
- the product of any matrix of type of N with any matrix of type A is a matrix of the type A;
- the product of any two matrices of type N is a matrix of type N.

Proposition 4. *Consider a Boolean sequence (c) of length $t+1$ and the Walsh matrices of the two (n,n)–functions f^0 and f^1: $W_{f^0} = A_{f^0} + N_{f^0}$ and $W_{f^1} = A_{f^1} + N_{f^1}$. The product $W = W_{f^{c_t}} \times \cdots \times W_{f^{c_0}}$ is of type A if and only if the matrix $N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$ is null.*

Proof.

$$\begin{aligned} W &= W_{f^{c_t}} \times \cdots \times W_{f^{c_0}} \\ &= (A_{f^{c_t}} + N_{f^{c_t}}) \times \cdots \times (A_{f^{c_0}} + N_{f^{c_0}}) \end{aligned} \tag{19}$$

By expanding the expression and using Remark 3, W can be rewritten $W = A + N$ with A a type A matrix and $N = N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$ a type N matrix. Because of its structure, A cannot cancel the non zero coefficients of N. Therefore, W is a type A matrix if and only if N is null.

The self-synchronization property in the spectral domain can have an algebraic interpretation. It is based on the concept of semigroup.

A semigroup is a set together with an associative multiplication. For instance the set of the $2^n \times 2^n$ Walsh matrices together with the multiplication defined by (14) is a semigroup. A nilpotent element e is an element such that there exists a large enough positive integer k such that $e^k = 0$. A semigroup is said to be generated by a family of elements $E = \{e_0, \ldots, e_n\}$ if any element of the semigroup can be expressed in terms of a product of finite length of elements of E. A nilpotent semigroup is a semigroup with a zero element and in which each element is nilpotent. The nilpotency class of a semigroup S is the smallest positive integer k such that $\forall e \in S$, $e^k = 0$.

Proposition 5. *The system (5)–(6) is finite time self-synchronizing if and only if the matrices N_{f^0} and N_{f^1} span a nilpotent semigroup.*

Proof. According to Remark 2, a system is finite time self-synchronizing if and only if there is a positive integer t_c such that any sequence of length greater than t_c is self-synchronizing. That is, in view of Proposition 2 and Parseval's Theorem (10), for $t > t_c$, any Walsh matrix $W_{\phi_t^c}$ is of type A. The expression of $W_{\phi_t^c}$ given by (17) is, up to a constant factor, the product of $t+1$ elements of the pair $\{W_{f^0}, W_{f^1}\}$. According to Proposition 4 this product is of type A if and only if whatever is $c \in \mathbb{F}_2^{t+1}$, the product $N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$ is null. This is the case if and only if $\{N_{f^0}, N_{f^1}\}$ spans a nilpotent semigroup of nilpotency class at most $t+1$.

Now, we aim at pinpointing different classes of self-synchronizing functions. To this end, let us recall an interesting theorem stated in [6] (Theorem 2.1.7).

Theorem 2 (Levitski's theorem). *Any semigroup of nilpotent matrices is triangularizable.*

For any square Walsh matrix W of dimension 2^n, let us define its reduced matrix W^* of dimension $(2^n-1) \times (2^n-1)$ which is the matrix W in which the first row and column have been removed.

$$W^* = \begin{pmatrix} w_{1,1} & \cdots & w_{1,q-1} \\ \vdots & & \vdots \\ w_{q-1,1} & \cdots & w_{q-1,q-1} \end{pmatrix}$$

Remark 4. Note that the reduced matrix of N is W^* as well.

Next proposition makes a classification of the possible situations that allow the system (5)–(6) to be finite time self-synchronizing. It clearly gives a characterization of the functions that can be used in the design of finite time SSSC.

Proposition 6. *The system* (5)–(6) *with the next-state function f (and the associated (n,n)–functions f^0 and f^1) is finite time self-synchronizing if and only if the reduced Walsh matrices $W^*_{f^0}$ and $W^*_{f^1}$ are nilpotent and fulfill one of the following cases:*

Case 1 Both matrices $W^*_{f^0}$ and $W^*_{f^1}$ are lower triangular;

Case 2 Both matrices $W^*_{f^0}$ and $W^*_{f^1}$ are not lower triangular but can be simultaneously triangularized by a change of basis whose matrix is the reduced Walsh matrix W^*_p of some (n,n)–function p. This matrix has to be invertible. In this situation, the following equalities hold: $W^*_p W^*_{f^0} (W^*_p)^{-1} = \widetilde{W}^*_{f^0}$ and $W^*_p W^*_{f^1} (W^*_p)^{-1} = \widetilde{W}^*_{f^1}$ with $\widetilde{W}^*_{f^0}$ and $\widetilde{W}^*_{f^1}$ two lower triangular matrices with zeros on the diagonal;

Case 3 Both matrices $W^*_{f^0}$ and $W^*_{f^1}$ are not lower triangular. They can be however simultaneously triangularized like in Case 2 but unlike Case 2, W^*_p does not correspond to a Walsh matrix.

Proof. Proposition 5 states that the system (5)–(6) is finite time self-synchronizing if and only if N^0_f and N^1_f span a nilpotent semigroup. In view of Remark 4, the same holds for the matrices $W^*_{f^0}$ and $W^*_{f^1}$. Then, in view of Theorem 2, they can be simultaneously triangularized. Cases 1, 2 and 3 are exclusive and describe all the possible situations.

Case 1 corresponds to the case when f^0 and f^1 are strict T–functions. Indeed, the reduced Walsh matrix is lower triangular with zeros on diagonal except on the first row if and only if the corresponding function is a strict T–function (see Proposition 11 [7]). Therefore Case 1 refers to functions which have been already proposed through the open literature.

Case 2 corresponds to the situation when f^0 and f^1 are not strict T–functions but functions of the form $f^0 = p \circ \widetilde{f}^0 \circ p^{-1}$ and $f^1 = p \circ \widetilde{f}^1 \circ p^{-1}$ where $\widetilde{f}^0$ and $\widetilde{f}^1$ are strict T–functions and p a bijection over $\mathbb{F}_2^n$. Indeed, since the invertible Walsh matrices are exactly the Walsh matrices of

the bijections over $\mathbb{F}_2^n$. Moreover, if p is a bijective (n,n)–function, $(W_p^*)^{-1} = W_{p^{-1}}^*$. Therefore, this case is nothing but Case 1 in which the functions f^0 and f^1 have been both left-composed with the same bijective function p and right composed with p^{-1}. Thus, this case is equivalent to Case 1 up to an invertible transformation of the internal states.

Case 3 corresponds to self-synchronizing functions that are not based on strict T–functions. This case is the most interesting one insofar as it defines new classes of self-synchronizing functions. An example of such a function is given in Section 6.

Remark 5. It is interesting to note that the synchronization delay t_c precisely corresponds to the nilpotency class of the semigroup spanned by $W_{f^0}^*$ and $W_{f^1}^*$. Moreover, since Cases 1 and 2 are based on strict T–functions, the maximum nilpotency class is bounded by n in these situations. In Case 3 the maximum nilpotency class is the dimension of the matrices which is $2^n - 1$. Therefore, if two reduced Walsh matrices $W_{f^0}^*$, $W_{f^1}^*$ span a nilpotent semigroup of nilpotency class greater than n, it necessary corresponds to Case 3.

The problem of determining if any two (n,n)–functions f^0 and f^1 can be used to design finite-time self-synchronizing systems as defined by (5)–(6) amounts to check whether or not their reduced Walsh matrices $W_{f^0}^*$ and $W_{f^1}^*$ span a nilpotent semigroup. From Proposition 5, if this is the case they can be simultaneously triangularized. The book [6] provides interesting approaches to determine whether or not a set of matrices can be simultaneous triangularized. An algorithm that simultaneously triangularizes a set of matrices is given in the paper [8]. The algorithm can be applied to any set of matrices, it simply fails when no common triangularization basis exists. In the next paragraph, an extension of the finite time self-synchronization is proposed. It is called statistical self-synchronization.

4.3 Statistical Self-Synchronization

In this paragraph, in order to enlarge the class of potential candidate functions, we relax the finite time self-synchronization constraint and extend Definition 4. Indeed, in practice, it is acceptable that the synchronization delay t_c is not bounded, but may be a random variable with a probability law that decreases to zero as time goes to infinity. In other words, the probability of being synchronized reaches one while the length of the stream (c) increases as illustrated by Figure 3. In order for this concept to be practical, clearly, the probability of being synchronized must be sufficiently close to one for some reasonable length. Such systems are called statistical SSSC. If (c) is a random sequence then, the synchronization delay t_c is a random variable. In such a case, it is denoted T_c.

Definition 6 (Statistical self-synchronization). *The system* (5)–(6) *is statistically self-synchronizing if* $\lim_{t\to+\infty} \Pr(T_c \leq t) = 1$. *The random variable* T_c *is called the random synchronization delay for the random sequence* (c).

Remark 6. It is interesting to note that if the probability of synchronization is one for some constant delay, Definition 6 reduces to Definition 4. Therefore, finite time self-synchronization is nothing but a special case of statistical self-synchronization.

Even though Remark 6 states that finite time self-synchronization is a special case of statistical self-synchronization, in general, by *statistical self-synchronization*, it is meant *statistical self-synchronization which is not finite time self-synchronization.*

We now focus on statistical self-synchronizing systems. According to Section 4.2, when the pair $\{W_{f^0}^*, W_{f^1}^*\}$ spans a nilpotent semigroup, the associated function f is finite time self-synchronizing. If the pair does not span a nilpotent semigroup, the synchronization delay cannot

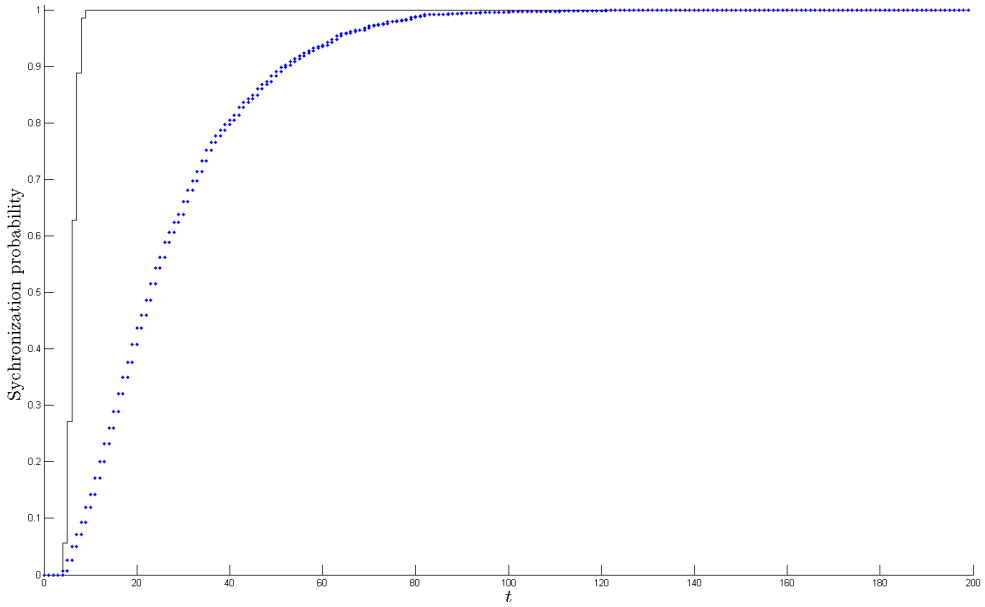

Fig. 3: Synchronization probability with respect to time. Solid line: finite-time self-synchronization. Dotted line: statistical self-synchronization

be bounded anymore since some sequences are not self-synchronizing. The probability of synchronization at time t is upper bounded by the probability that a self-synchronizing sequence appears for the first time at time t in the stream (c) since some non self-synchronizing sequences might synchronize the system for some specific pairs of initial states x_0 and $\widehat{x}_0$. Note that, determining the probability that a specific sequence appears for the first time in a uniform random sequence of a given length is not trivial. This is mainly due to the fact that two different sequences even with equal length do not necessary have the same probability of appearing in a uniform random sequence. However, when considering all the self-synchronizing sequences of a specific length, we can resort to the result provided in the paper [9].

5 State Probability

Ensuring the self-synchronizing property is a first feature required for the design of SSSC. The security has to be further assessed. From this perspective, it should be interesting to determine the probability that a given state can be reached after a fixed number of iterations. Clearly, all the states have to be reachable and with almost the same probability (ideally the same). It is the purpose of this section.

Let us first consider the following matter. Given a (n, m)–function g and a vector $x \in \mathbb{F}_2^n$ whose value is described by the probability law $p : \mathbb{F}_2^n \longrightarrow \mathbb{R}$, express the probability law of $y = g(x) \in \mathbb{F}_2^m$ defined by $q : \mathbb{F}_2^m \longrightarrow \mathbb{R}$. For the $\mathbb{F}_2^n$ valued random variable X, the function p is defined by $p(x) = \Pr[X = x]$ and the function q is defined by $q(y) = \Pr[g(X) = y]$. Without ambiguity, the notation p (respectively q) refers either to the function or to the 2^n (respectively 2^m) column vector whose component index $x \in \mathbb{F}_2^n$ (respectively $y \in \mathbb{F}_2^m$) has the value $p(x)$ (respectively $q(y)$). The same holds for $\widehat{p}$ and $\widehat{q}$ which are the Fourier transforms of p and q.

Proposition 7. *Let W_g be the Walsh matrix of g. Applying the function g to a variable whose value is chosen according to the probability law described by p gives a vector whose value is*

described by the probability law q. They are related by the relation

$$\widehat{q} = \frac{1}{2^n} W_g \widehat{p} \tag{20}$$

Proof. Let us first relate q and p.

$$\begin{aligned} q(y) &= \textstyle\sum_{x\in\mathbb{F}_2^n | g(x)=y} p(x) \\ &= 2^{-m} \textstyle\sum_{x\in\mathbb{F}_2^n} p(x) \sum_{u\in\mathbb{F}_2^m} (-1)^{u(g(x)+y)} \\ &= 2^{-m} \textstyle\sum_{u\in\mathbb{F}_2^m} \sum_{x\in\mathbb{F}_2^n} (-1)^{u\cdot y} p(x) (-1)^{u\cdot g(x)} \end{aligned}$$

We now express the Fourier transform of q.

$$\begin{aligned} \widehat{q}(s) &= \textstyle\sum_{y\in\mathbb{F}_2^m} q(y)(-1)^{s\cdot y} \\ &= 2^{-m} \textstyle\sum_{u\in\mathbb{F}_2^m, x\in\mathbb{F}_2^n} \underbrace{\sum_{y\in\mathbb{F}_2^m} (-1)^{u\cdot y + s\cdot y}}_{\begin{cases} 2^m & \text{if } u = s \\ 0 & \text{else} \end{cases}} p(x)(-1)^{u\cdot g(x)} \\ &= \textstyle\sum_{x\in\mathbb{F}_2^n} p(x)(-1)^{s\cdot g(x)} \\ &= 2^{-n} \textstyle\sum_{x\in\mathbb{F}_2^n} p(x) \sum_{z\in\mathbb{F}_2^n} (-1)^{s\cdot g(z)} \sum_{v\in\mathbb{F}_2^n} (-1)^{v\cdot(x+z)} \\ &= 2^{-n} \textstyle\sum_{v\in\mathbb{F}_2^n} \underbrace{\sum_{x\in\mathbb{F}_2^n} p(x)(-1)^{v\cdot x}}_{\widehat{p}(v)} \underbrace{\sum_{z\in\mathbb{F}_2^n} (-1)^{s\cdot g(z) + v\cdot z}}_{w^g_{s,v}} \end{aligned}$$

where $w^g_{s,v}$ is as defined by (13). The result holds.

The following corollary can be stated

Corollary 1. *Let X follows the uniform distribution and g be a (n,n)–function. The probability distribution, after applying the function g to X reads*

$$\forall x \in \mathbb{F}_2^n,\ \Pr[g(X) = x] = 2^{-2n} \sum_{s\in\mathbb{F}_2^n} (-1)^{s\cdot x} w^g_{s,0} \tag{21}$$

where $w^g_{s,0}$ is the coefficient of the Walsh matrix of g of the s^{th} row and of the first column.

Proof. If we assume a uniform distribution of the initial state, p is the constant function $\forall x \in \mathbb{F}_2^n, p(x) = 2^{-n}$. Its Fourier transform $\widehat{p}$ is

$$\widehat{p}(u) = \begin{cases} 1 & \text{if } u = 0 \\ 0 & \text{otherwise} \end{cases}$$

From Proposition 7, the equality $\widehat{q}(u) = w^g_{u,0}$ holds. Using the inverse Fourier transform formula (9) completes the proof.

It could be interesting to relate Proposition 7 and Corollary 1 to [10] since they extend Lemma 1 of the latter paper. They are more general in the sens that we do not consider any specific probability distribution.

Till now, we have always considered that the initial state x_0 is chosen according to the uniform probability. We have also assumed that the symbols of the ciphertext stream (c) are uniformly

distributed. Let us stress that even though this assumption makes sense in cryptography since the stream (c) should not be distinguishable from a true uniform random stream, it should be considered with caution. Indeed, the uniformity of (c) depends on the uniformity of (z) which in turn depends on the function f and h.

We now focus on the evolution of the probability law modified by the next-state function f.

Proposition 8. *Let (C) be a uniform random sequence and assume a uniform random distribution of the initial state X_0. Then, the probability that the iterated function ϕ_t returns the state $x \in \mathbb{F}_2^n$ is*

$$P[\phi_t^C(X_0) = x] = \frac{1}{2^{2n+n\cdot t+t+1}} \sum_{s\in\mathbb{F}_2^n} (-1)^{s\cdot x} \left[\left[W_{f^0} + W_{f^1} \right]^{t+1} \right]_{s,0} \tag{22}$$

Proof. Since (C) is a uniform random sequence of length $t+1$, the probability of having this specific sequence in $t+1$ iterations is 2^{-t-1}.

$$P[\phi_t^C = x] = \frac{1}{2^{t+1}} \sum_{c\in\mathbb{F}_2^{t+1}} P[f^{C_k} \circ \cdots \circ f^{C_0} = x]$$

Then, in view of Proposition 1 and Corollary 1

$$\begin{aligned} P[\phi_t^C = x] &= \frac{1}{2^{t+1}} \sum_{c\in\mathbb{F}_2^{t+1}} \frac{1}{2^{2n}} \sum_{s\in\mathbb{F}_2^n} (-1)^{s\cdot x} \frac{1}{2^{n\cdot t}} \left[W_{f^{C_k}} \times \cdots \times W_{f^{C_0}} \right]_{s,0} \\ &= \frac{1}{2^{2n+n\cdot t+t+1}} \sum_{s\in\mathbb{F}_2^n} (-1)^{s\cdot x} \left[\left[W_{f^0} + W_{f^1} \right]^{t+1} \right]_{s,0} \end{aligned}$$

Proposition 9. *Assuming a random sequence (C) of length $t+1$ and a random initial state X_0, the system (5)–(6) has an equal probability to be in each state if and only if the first column of the matrix $\left[W_{f^0} + W_{f^1}\right]^{t+1}$ denoted w_0 is given by*

$$w_0 = \left(2^{(n+1)(t+1)}\ 0 \cdots 0\right)^T \tag{23}$$

Proof. Proving this result amounts to solving a linear algebra problem. Let ν be the 2^n–dimensional column vector whose coefficients at row x is the probability of being in the state x. In our case, we set the value of each coefficient to 2^{-n}. Considering Proposition 8, denoting by H the 2^n–dimensional Hadamard matrix defined by $H = (h_{s,x}) = (-1)^{s\cdot x}$ for $s, x \in \mathbb{F}_2^n$ and by k the constant $2^{-2n-n\cdot t-t-1}$, the problem can be written

$$\nu = kHw_0$$

where w_0 is the unknown. Since both k and H are invertible, the system can be solved and has a unique solution.

Remark 7. The fact that each state is reached with an equal probability for a random sequence of length $t+1$ does not mean that each state is reached with an equal probability with a uniform random sequence of length $t+2$.

Remark 8. Proposition 9 states that assuming a uniform distribution of the initial state X_0 and a uniform random sequence (C), the uniform distribution of the internal state at time t is achieved if and only if the first column vector of $\left[W_{f^0} + W_{f^1}\right]^{t+1}$ is given by the relation (23). Under this condition, a uniform distribution is achieved at any time if and only if f is balanced.

Since the first column of $W_{f^0} + W_{f^1}$ is the same as the first column of W_f, if f is balanced, the condition $w_{s,0}^{f^0} = -w_{s,0}^{f^1}$ if $s \neq 0$ holds.

6 Example

As an illustration of the finite time SSSC described by Case 3 in Section 4.2, let us show that the set of functions described by Case 3 is not empty. We consider $n = 3$. The next-state function f is described by its restrictions f^0 and f^1 as defined by (15):

$$\begin{cases} f_0^0(x) = x_1 + x_0x_1 + x_2 + x_0x_2 \\ f_1^0(x) = x_1 + x_0x_1 + x_0x_2 + x_1x_2 + x_0x_1x_2 \\ f_2^0(x) = x_2 + x_0x_2 \end{cases} \qquad \begin{cases} f_0^1(x) = x_0x_1 + x_0x_2 + x_1x_2 \\ f_1^1(x) = x_2 + x_0x_1x_2 \\ f_2^1(x) = x_1x_2 \end{cases}$$

Then, the reduced Walsh matrices can be worked out by using (13)

$$W_{f^0}^* = \begin{pmatrix} 4 & 0 & 0 & -4 & 4 & 0 & 0 \\ 2 & 2 & 2 & 2 & -6 & 2 & 2 \\ 6 & 2 & 2 & -2 & -2 & 2 & 2 \\ 0 & 0 & 4 & -4 & 0 & 0 & 4 \\ 0 & 4 & 0 & -4 & 0 & 4 & 0 \\ 6 & -2 & 2 & 2 & -2 & -2 & 2 \\ 6 & 2 & -2 & 2 & -2 & 2 & -2 \end{pmatrix} \quad W_{f^1}^* = \begin{pmatrix} 4 & 4 & -4 & 0 & 0 & 0 & 0 \\ 6 & -2 & 2 & -2 & 2 & 2 & -2 \\ 6 & -2 & 2 & 2 & -2 & -2 & 2 \\ 4 & 4 & 0 & 4 & 0 & 0 & -4 \\ 0 & 0 & 4 & 4 & 0 & 0 & -4 \\ 2 & 2 & 2 & 2 & 2 & -6 & 2 \\ 2 & 2 & 2 & -2 & 6 & -2 & -2 \end{pmatrix}$$

According to Theorem 2, the matrices $W_{f^0}^*$ and $W_{f^1}^*$ span a nilpotent semigroup. Indeed, they can be simultaneously triangularized and the algorithm of the paper [8] allows to find out one possible change of basis. The matrix

$$W_p^* = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

triangularizes both $W_{f^0}^*$ and $W_{f^1}^*$.

It can be checked that the class of nilpotency of this semigroup is $\mathcal{C} = 4$. Since $\mathcal{C} > n$ and because of Remark 5 there is no bijection from $\mathbb{F}_2^3$ that allows to triangularize the semigroup. Therefore this system correspond to Case 3.

7 Conclusion

Two kinds of self-synchronization have been defined. Finite time self-synchronization has been characterized from the spectral analysis point of view. It has been shown that it is possible to achieve finite time self-synchronization using functions which are not strict T–functions. Three cases have been pinpointed. The known strict T–function case, the case when strict T–functions have been left and right composed with a permutation and its inverse and the case which is not based on strict T–functions. The latter case is interesting due to its novelty, algebraic characterization in terms of nilpotent semi-groups has been performed. We have then discussed statistical self-synchronization as a generalization of finite time self-synchronization.

These characterizations will prove constructive to the task of finding classes of keyed families functions for cryptographic purposes. A deeper insight is required to fully specify self-synchronizing stream ciphers so as to achieve security in the context of performance constraints.

Acknowledgment

This work was partially supported by the PEPS project AutoCrypt, Institut des Sciences et de l'Ingénierie des Systèmes, CNRS

References

1. J. Daemen. *Cipher and Hash function design, strategies based on linear and differential cryptanalysis.* PhD Thesis, Katholieke Universiteit Leuven, 1995.
2. A. Joux and F. Muller. Chosen-ciphertext attacks against mosquito. *Fast Software Encryption, Lecture Note in Computer Science*, 4047:87–99, Springer 2006.
3. Alexander Klimov and Adi Shamir. A new class of invertible mappings. In Burton Kaliski, etin Ko, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 470–483. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36400-5_34.
4. C. Carlet. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Boolean Functions for Cryptography and Error-Correcting Codes. In [12], 2010.
5. C. Carlet. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Vectorial Boolean Functions for Cryptography. In [12], 2010.
6. H. Radjavi and P. Rosenthal. *Simultaneous Triangularization.* Springer, 2000.
7. Jérémy Parriaux, Philippe Guillot, and Gilles Millérioux. Synchronization of boolean dynamical systems: A spectral characterization. In Claude Carlet and Alexander Pott, editors, *Sequences and Their Applications SETA 2010*, volume 6338 of *Lecture Notes in Computer Science*, pages 373–386. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15874-2_32.
8. C. Dubi. An algorithmic approach to simultaneous triangularization. *Linear Algebra and its Applications*, 430(11-12):2975 – 2981, 2009.
9. Dragana Bajic and Cedomir Stefanovic. Statistical analysis of search for set of sequences in random and framed data. In Claude Carlet and Alexander Pott, editors, *Sequences and Their Applications SETA 2010*, volume 6338 of *Lecture Notes in Computer Science*, pages 320–332. Springer Berlin / Heidelberg, 2010.
10. Kaisa Nyberg and Miia Hermelin. Multidimensional Walsh Transform and a Characterization of Bent Functions. In P. Vijay Kumar Tor Helleseth and Oyvind Ytrehus, editors, *Proceedings of the 2007 IEEE Information Theory Workshop on Information Theory for Wireless Networks*, IEEE, pages 83–86, 2007.
11. J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *SFCS '88: Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, Washington, DC, USA, 1988. IEEE Computer Society.
12. Y. Crama. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering.* Cambridge Press, 2010.
13. G. Millérioux and P. Guillot. Self-synchronizing stream ciphers and dynamical systems: state of the art and open issues. *International Journal of Bifurcation and Chaos, 20(9)*, 2010.
14. Nathan Keller. On the influence of variables on boolean functions in product spaces, May 2009. Availaible at: `http://arxiv.org/PS_cache/arxiv/pdf/0905/0905.4216v1.pdf`.
15. Alexander Klimov and Adi Shamir. Cryptographic applications of t-functions. In Mitsuru Matsui and Robert Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-24654-1_18.